Package ‘princurve’

May 29, 2019

Version  2.1.4

Title  Fits a Principal Curve in Arbitary Dimension

Description  Fitting a principal curve to a data matrix in arbitrary dimensions.

License  GPL-2

Encoding  UTF-8

Depends  R (>= 3.0)

Imports  stats, graphics, grDevices, Rcpp

Suggests  devtools, testthat

LinkingTo  Rcpp

NeedsCompilation  yes

RoxygenNote  6.1.1

URL  https://github.com/rcannood/princurve

BugReports  https://github.com/rcannood/princurve/issues

Collate  'RcppExports.R' 'bias_correct_curve.R' 'deprecated.R'
'periodic_lowess.R' 'smoother_functions.R'
'principal_curve.R' 'start_circle.R'

Author  Trevor Hastie [aut],
Andreas Weingessel [aut],
Kurt Hornik [aut] (<https://orcid.org/0000-0003-4198-9911>),
Henrik Bengtsson [ctb] (HenrikBengtsson),
Robrecht Cannoodt [aut, cre] (<https://orcid.org/0000-0003-3641-729X>,
rcannood)

Maintainer  Robrecht Cannoodt <rcannood@gmail.com>

Repository  CRAN

Date/Publication  2019-05-29 14:20:02 UTC
R topics documented:

princurve-package .................................................. 2
principal.curve ..................................................... 2
principal_curve ..................................................... 3
project_to_curve ................................................... 5
smoother_functions .................................................. 6
start_circle ............................................................ 6

Index

princurve-package Fits a Principal Curve in Arbitrary Dimension

Description

Fits a Principal Curve in Arbitrary Dimension

References

See also Banfield and Raftery (JASA, 1992).

See Also

principal_curve, project_to_curve

principal.curve Deprecated functions

Description

This function is deprecated, please use principal_curve and project_to_curve instead.

Usage

principal.curve(x, start = NULL, thresh = 0.001, plot.true = FALSE,
maxit = 10, stretch = 2, smoother = c("smooth_spline", "lowess",
"periodic_lowess"), trace = FALSE, ...)

## S3 method for class 'principal.curve'
lines(x, ...)

## S3 method for class 'principal.curve'
plot(x, ...)
# principal_curve

## S3 method for class 'principal.curve'

points(x, ...)

get.lam(x, s, tag = NULL, stretch = 2)

### Arguments

- **x**: x
- **start**: start
- **thresh**: thresh
- **plot.true**: plot.true
- **maxit**: maxit
- **stretch**: stretch
- **smoother**: smoother
- **trace**: trace
- **...**: ...
- **s**: s
- **tag**: tag

---

## Description

Fits a principal curve which describes a smooth curve that passes through the middle of the data x in an orthogonal sense. This curve is a nonparametric generalization of a linear principal component. If a closed curve is fit (using smoother = "periodic_lowess") then the starting curve defaults to a circle, and each fit is followed by a bias correction suggested by Jeff Banfield.

### Usage

```r
call = principal_curve(x, start = NULL, thresh = 0.001, maxit = 10, stretch = 2, smoother = c("smooth_spline", "lowess", "periodic_lowess"), approx_points = FALSE, trace = FALSE, plot_iterations = FALSE, ...)
```

## S3 method for class 'principal.curve'

lines(x, ...)

## S3 method for class 'principal.curve'

plot(x, ...)

## S3 method for class 'principal.curve'

points(x, ...)

whiskers(x, s, ...)
Arguments

- **x**: A matrix of points in arbitrary dimension.
- **start**: Either a previously fit principal curve, or else a matrix of points that in row order define a starting curve. If missing or NULL, then the first principal component is used. If the smoother is "periodic_lowess", then a circle is used as the start.
- **thresh**: Convergence threshold on shortest distances to the curve.
- **maxit**: Maximum number of iterations.
- **stretch**: A stretch factor for the endpoints of the curve, allowing the curve to grow to avoid bunching at the end. Must be a numeric value between 0 and 2.
- **smoother**: Choice of smoother. The default is "smooth_spline", and other choices are "lowess" and "periodic_lowess". The latter allows one to fit closed curves. Beware, you may want to use iter = 0 with lowess().
- **approx_points**: Approximate curve after smoothing to reduce computational time. If FALSE, no approximation of the curve occurs. Otherwise, approx_points must be equal to the number of points the curve gets approximated to; preferably about 100.
- **trace**: If TRUE, the iteration information is printed.
- **plot_iterations**: If TRUE the iterations are plotted.
- **...**: Additional arguments to the smoothers.
- **s**: A parametrized curve, represented by a polygon.

Value

An object of class "principal_curve" is returned. For this object the following generic methods a currently available: `plot`, `points`, `lines`.

It has components:

- **s**: A matrix corresponding to x, giving their projections onto the curve.
- **ord**: An index, such that s[order,] is smooth.
- **lambda**: For each point, its arc-length from the beginning of the curve. The curve is parametrized approximately by arc-length, and hence is unit-speed.
- **dist**: The sum-of-squared distances from the points to their projections.
- **converged**: A logical indicating whether the algorithm converged or not.
- **num_iterations**: Number of iterations completed before returning.
- **call**: The call that created this object; allows it to be updated().

References


See Also

- `project_to_curve`
Examples

```r
x <- runif(100, -1, 1)
x <- cbind(x, x^2 + rnorm(100, sd = 0.1))
fit <- principal_curve(x)
plot(fit)
lines(fit)
points(fit)
whiskers(x, fit$s)
```

---

**project_to_curve**  
*Project a set of points to the closest point on a curve*

**Description**

Finds the projection index for a matrix of points `x`, when projected onto a curve `s`. The curve need not be of the same length as the number of points.

**Usage**

```r
project_to_curve(x, s, stretch = 2)
```

**Arguments**

- `x`: a matrix of data points.
- `s`: a parametrized curve, represented by a polygon.
- `stretch`: A stretch factor for the endpoints of the curve, allowing the curve to grow to avoid bunching at the end. Must be a numeric value between 0 and 2.

**Value**

A structure is returned which represents a fitted curve. It has components

- `s`: The fitted points on the curve corresponding to each point `x`
- `ord`: the order of the fitted points
- `lambda`: The projection index for each point
- `dist`: The total squared distance from the curve
- `dist_ind`: The squared distances from the curve to each of the respective points

**See Also**

- `principal_curve`
**smoother_functions**  
*Smoother functions*

**Description**

Each of these functions have an interface `function(lambda, xj, ...)`, and return smoothed values for `xj`. The output is expected to be ordered along an ordered lambda. This means that the following is true:

```r
x <- runif(100)
y <- runif(100)
ord <- sample.int(100)
sfun <- smoother_functions[[1]]
all(sfun(x, y) == sfun(x[ord], y[ord]))
```

**Usage**

`smoother_functions`

**Format**

An object of class `list` of length 3.

---

**start_circle**  
*Generate circle as initial curve*

**Description**

The starting circle is defined in the first two dimensions, and has zero values in all other dimensions.

**Usage**

`start_circle(x)`

**Arguments**

- `x`  
The data for which to generate the initial circle
Examples

```r
## Not run:
x <- cbind(
  rnorm(100, 1, .2),
  rnorm(100, -5, .2),
  runif(100, 1.9, 2.1),
  runif(100, 2.9, 3.1)
)
circ <- start_circle(x)
plot(x)
lines(circ)
## End(Not run)
```
Index

*Topic **datasets**
  smoother_functions, 6

*Topic **nonparametric**
  principal_curve, 3
  princurve-package, 2
  project_to_curve, 5

*Topic **regression**
  principal_curve, 3
  princurve-package, 2
  project_to_curve, 5

*Topic **smooth**
  principal_curve, 3
  princurve-package, 2
  project_to_curve, 5

get.lam(principal.curve), 2

lines.principal.curve
  (principal.curve), 2
lines.principal.curve
  (principal.curve), 3

plot.principal.curve (principal.curve),
  2
plot.principal.curve (principal.curve),
  3
points.principal.curve
  (principal.curve), 2
points.principal.curve
  (principal.curve), 3
principal.curve, 2
principal_curve, 2, 3, 5
princurve (princurve-package), 2
princurve-package, 2
project_to_curve, 2, 4, 5

smoother_functions, 6
start_circle, 6

whiskers (principal_curve), 3